

# 第一次实验实验报告

李浩然 2025 年 10 月 29 日

## 1 实验概述

### 1.1 实验目的

本实验基于 LC-3 处理器架构, 通过机器码编程实现 C 语言中三目运算符 ( $R0 = (R0 \neq 0) ? R1 : R2$ ) 的功能。旨在掌握 LC-3 指令集的基本使用方法, 理解条件码寄存器 (N/Z/P) 的作用机制, 以及分支跳转指令在逻辑控制中的应用, 同时熟悉机器码程序的编写、调试流程。

### 1.2 实验要求

根据实验文档要求, 需满足以下约束:

- 初始条件: 寄存器 R0 存放判断条件 (0 或 1), R1 存放条件成立时的值 (val1), R2 存放条件不成立时的值 (val2)
- 功能目标: 程序执行后, R0 需存储正确结果 (R0=1 时存 R1 的值, R0=0 时存 R2 的值)
- 程序规范: 必须从内存地址 x3000 开始执行, 以 HALT 指令 (1111000000100101) 终止, 指令总数不超过 100 条, 且不得访问内存 (仅使用寄存器操作)

## 2 实验设计与实现

### 2.1 核心思路

采用“条件判断 + 分支跳转”的方案, 步骤如下:

- 利用 ADD 指令触发条件码更新, 让处理器记录 R0 的当前状态 (0 或 1)
- 使用 BR 指令根据条件码进行分支选择: 若 R0 为 1 (正数, P 条件码置 1), 跳转到“R0=R1”的执行逻辑; 若 R0 为 0 (零, Z 条件码置 1), 则执行“R0=R2”的逻辑

### 2.2 机器码实现

最终实现的机器码如下 (共 7 条指令, 满足所有约束):

Listing 1: 源码

1	0011000000000000	; x3000: 程序起始地址标记 (必须保留)
2	0001000000100000	; x3001: ADD R0, R0, #0 (更新条件码, 记录R0状态)
3	0000001000000010	; x3002: BRp 偏移+2 (若R0=1, 跳转到x3005执行R0=R1)
4	0001000010100000	; x3003: ADD R0, R2, #0 (R0=0时, 暂存R2的值)
5	0001001010100000	; x3004: ADD R1, R2, #0 (将R2的值暂存到R1, 防止覆盖)
6	0001000001100000	; x3005: ADD R0, R1, #0 (最终结果存入R0)
7	1111000000100101	; x3006: HALT (程序终止指令)

### 3 测试与结果分析

#### 3.1 测试用例

设计两组典型测试用例验证程序正确性：

表 1: 实验测试用例

测试用例	R0 初始值 (cond)	R1 (val1)	R2 (val2)	预期 R0 结果
1	0	xb91d	xe4a7	xe4a7
2	1	xd6a2	xc628	xd6a2

#### 3.2 测试结果

所有测试用例均通过，程序功能正确。

### 4 实验总结

#### 4.1 遇到的问题与解决方案

1. 跳转偏移计算错误：初期 BR 指令偏移量设置错误，通过“目标地址 = 当前 PC+1+ 偏移量”公式重新计算修正
2. 修正了 BR 不等于“if”的特性

#### 4.2 实验收获

1. 掌握了 LC-3 指令集中 ADD 和 BR 指令的使用方法，理解了条件码寄存器的作用
2. 学会了机器码程序的一些调试技巧
3. 深入理解了底层程序如何实现高层语言中的逻辑运算（如三目运算符）

#### 4.3 改进方向

可尝试实验文档中提到的“无分支解决方案” (Branchless Solution)，通过位运算实现条件选择，减少跳转指令带来的性能开销。